

1 **DATA PROCESSING SYSTEMS**

2 **FIELD OF INVENTION**

3 The present invention generally relates to data processing
4 systems. It particularly relates to security in data
5 processing systems, and especially to controlling access to
6 resources in data processing systems.

7 **BACKGROUND OF THE INVENTION**

8 For a general overview of security in data processing, see,
9 for example, Simone Fischer-Huebner: IT-Security and
10 Privacy, 2001 and Dorothy Denning: Cryptography and Data
11 Security, 1982. An aspect of security in the data processing
12 field is that of controlled access to objects or resources
13 such as data files and the like. Such access control is
14 typically implemented with reference to attributes of a user
15 seeking access. The attributes might include, for example,
16 subscription status, or clearance to read or write sensitive
17 data. A data processing process in which performance of the
18 process is dependent on one or more attributes of a user
19 seeking to perform the process is typically referred to as a
20 task. Examples of such tasks include reading from and
21 writing to a classified data file.

22 In M. Abrams, J. Heaney, O. King, L. LaPadula, M. Lazear, I.
23 Olson: Generalized Framework for Access Control: Towards
24 Prototyping the ORGCON Policy, In Proceedings of the 14th
25 National Computer Security Conference, Baltimore, October

1 1991, there is described a Generalized Framework for Access
2 Control (GFAC) as shown in Figure 1. The GFAC is typically
3 implemented in software to implement one or more access
4 control schemes in a data processing system comprising a
5 central processing unit (CPU), memory subsystem, and
6 input/output (I/O) subsystem all interconnected via a bus
7 subsystem. The GFAC is typically stored in the memory for
8 execution by the CPU.

9 Referring to Figure 1, the GFAC comprises an Access Control
10 Enforcement Facility (AEF) 10. The AEF 10 resides in a
11 Trusted Computing Base (TCB) 20. The TCB 20 is a protected
12 part of the data processing system, such as an operating
13 system kernel. In operation, the AEF 10 receives an access
14 request 30 from a subject 40. The subject 40 is typically
15 manifested by its proxy. The proxy is a task which inherits
16 access rights from the requesting subject 40. The subject 40
17 might for example be a user having defined access rights.
18 Such access rights might include the right to read from a
19 file or the right to write to a file. Access functions such
20 as reading and writing may be regarded as having different
21 sensitivities. For example, there may be more risk
22 associated with a write operation to a file than with a read
23 operation. In use, the AEF 10 blocks or grants requests 30
24 for access 100 to an object 110, such as a classified data
25 file. However, the AEF 10 delegates decision making to an
26 Access Control Decision Facility (ADF) 50. Specifically, on
27 receipt of the request 30, the AEF 10 sends the ADF 50 a
28 decision request 80. In response to the decision request 80,
29 the ADF 50 generates a decision 90 indicating whether it has
30 decided to grant or to deny the request 30. The ADF 50
31 refers to stored Access Control Information (ACI) 60 and
32 stored Access Control Rules (ACR) 70 to make its decision.

1 The ACI 60 comprises the attributes of the subject 40 and
2 the object 110. The ACR 70 comprises a set of rules defining
3 whether or not access to a given object can be granted to
4 the subject 40 based on the attributes of the subject 40. In
5 dependence on the decision 90 received from the ADF 50, the
6 AEF 10 either grants or denies the subject 40 access 100 to
7 the object 110. For simple privacy and security policies,
8 the decision process can be performed quickly. However, more
9 computation is needed when the ACR 70 specifies more
10 complicated rules. Accordingly, the decision may be delayed,
11 thus limiting system performance. Furthermore, some rules
12 may require knowledge of prior accesses to make a decision.
13 This brings additional delay and complicates implementation
14 of the GFAC. It would be desirable to avoid such delays and
15 complexity.

16 SUMMARY OF THE INVENTION

17 Therefore, in one aspect the present invention provides
18 methods, apparatus and systems for controlling access to an
19 object in a data processing system. An example method
20 comprising: receiving a request to access the object from a
21 task; classifying the access request into one of critical
22 and non-critical classes in dependence on stored access
23 control data associated with the object and the task;
24 granting the task access to the object and storing data
25 indicative of the access in an access log if the access is
26 classified into the non-critical class; and, in the event
27 that the access is classified into the critical class,
28 granting or denying the task access to the object in

1 dependence on the contents of the access log and the stored
2 access control data.

3 Preferably, the method comprises, in the event that the
4 access is classified into the non-critical class, granting
5 or denying the task access to the object in dependence on
6 the access control data, and storing data indicative of the
7 grant or denial in the access log.

8 Viewing the present invention from another aspect, there is
9 now provided apparatus for controlling access to an object
10 in a data processing system, the apparatus comprising: an
11 access control data store for storing access control data
12 associated with the object and the task; an access log;
13 access control logic for receiving a request to access the
14 object from a task; decision classifier logic, connected to
15 the access control logic, the access control data store, and
16 the access log, for classifying the access request into one
17 of critical and non-critical classes in dependence on the
18 access control data, and, in the event that the access is
19 classified into the non-critical class, for granting the
20 task access to the object and storing data indicative of the
21 access in the access log; and, access control decision logic
22 connected to the access control logic, the access log, the
23 access control data store, and the decision classifier
24 logic, for, in the event that the access is classified into
25 the critical class, granting or denying the task access to
26 the object in dependence on the contents of the access log
27 and the access control data. The present invention extends
28 to a data processing system comprising: a central processor
29 unit; a memory; and access control apparatus as herein
30 before described connected to the central processor unit and
31 the memory.

1 The present invention is particularly although not
2 exclusively applicable to privacy and data protection. For
3 example, consider a process that accesses, processes, and
4 discloses personal information. To enforce external privacy
5 policy, such disclosures are marked towards outsiders as
6 needing an immediate access control decision. For others,
7 deferred access control might be sufficient. This does not
8 prevent privacy violations within an enterprise, but it
9 prevents such privacy violations producing illegal
10 disclosures of personal information to outsiders.

11 **BRIEF DESCRIPTION OF THE DRAWINGS**

12 The invention and its embodiments will be more fully
13 appreciated by reference to the following detailed
14 description of advantageous and illustrative embodiments in
15 accordance with the present invention when taken in
16 conjunction with the accompanying drawings, in which:

17 Figure 1 is a block diagram of a Generalized Framework for
18 Access Control (GFAC);

19 Figure 2 is a block diagram of a data processing system;

20 Figure 3 is a logical block diagram of an example of access
21 control system embodying the present invention;

22 Figure 4 is a flow chart associated with the access control
23 system shown in Figure 3;

1 Figure 5 is another flow chart associated with the access
2 control system shown in Figure 3;

3 Figure 6 is a more detailed logical block diagram of the
4 access control system shown in Figure 3;

5 Figure 7 is a logical block diagram of another example of
6 access control system embodying the present invention;

7 Figure 8 is a flow diagram representative of multiple tasks
8 executing in a data processing system;

9 Figure 9 is a flow chart associated with the access control
10 system shown in Figure 7;

11 Figure 10 is another flow chart associated with the access
12 control system shown in Figure 7;

13 Figure 11 is a further flow chart associated with the access
14 control system shown in Figure 7; and,

15 Figure 12 is yet another flow chart associated with the
16 access control system shown in Figure 7.

17 **DETAILED DESCRIPTION OF THE INVENTION**

18 The present invention provides methods, systems and
19 apparatus for controlling access to an object in a data
20 processing system. In an example embodiment, a method
21 comprises: receiving a request to access the object from a
22 task; classifying the access request into one of critical
23 and non-critical classes in dependence on stored access

1 control data associated with the object and the task;
2 granting the task access to the object and storing data
3 indicative of the access in an access log if the access is
4 classified into the non-critical class; and, in the event
5 that the access is classified into the critical class,
6 granting or denying the task access to the object in
7 dependence on the contents of the access log and the stored
8 access control data.

9 Preferably, the method comprises, in the event that the
10 access is classified into the non-critical class, granting
11 or denying the task access to the object in dependence on
12 the access control data, and storing data indicative of the
13 grant or denial in the access log.

14 The non-critical class may comprise a plurality of
15 subclasses and the classifying may comprise classifying the
16 access request into one of the subclasses in dependence on
17 the stored access control data. In a preferred embodiment of
18 the present invention, the subclasses comprise a first
19 subclass and a second subclass. In a particularly preferred
20 embodiment of the present invention, recovery data is stored
21 in the access log if the access is classified into the
22 second subclass. The access log may be inspected to identify
23 bad grant decision based on the contents of the access log
24 and the access control data and the method may comprise, on
25 detection of a bad grant decision, rolling back any objects
26 affected by the bad grant decision. The rolling back may
27 comprise recovering data overwritten in the object. The
28 inspection may be performed periodically. Alternatively, the
29 inspecting may be performed during periods in which the data
30 processing system is otherwise idle.

1 There is now also provided apparatus for controlling access
2 to an object in a data processing system, the apparatus
3 comprising: an access control data store for storing access
4 control data associated with the object and the task; an
5 access log; access control logic for receiving a request to
6 access the object from a task; decision classifier logic,
7 connected to the access control logic, the access control
8 data store, and the access log, for classifying the access
9 request into one of critical and non-critical classes in
10 dependence on the access control data, and, in the event
11 that the access is classified into the non-critical class,
12 for granting the task access to the object and storing data
13 indicative of the access in the access log; and, access
14 control decision logic connected to the access control
15 logic, the access log, the access control data store, and
16 the decision classifier logic, for, in the event that the
17 access is classified into the critical class, granting or
18 denying the task access to the object in dependence on the
19 contents of the access log and the access control data. The
20 present invention extends to a data processing system
21 comprising: a central processor unit; a memory; and access
22 control apparatus as herein before described connected to
23 the central processor unit and the memory.

24 The present invention also extends to a computer program
25 element comprising computer program code means which, when
26 loaded in a processor of a computer system, configures the
27 processor to perform an access control method as herein
28 before described.

29 As will be appreciated from the following detailed
30 description of various embodiments of the present invention,
31 the decision classifier logic acts as a coarse filter of

1 decision requests. The access control decision logic
2 subsequently acts as a fine filter of those decision
3 requests passed to it via the decision triager.

4 By way of illustration of an advantage of the present
5 invention, consider a computational process P desiring
6 access to a secure object O, such as a stored data file, for
7 which permission to access is needed. Permission might be
8 granted in real time immediately before access is desired,
9 as herein before described with reference to the
10 conventional GFAC system. However, in general, checking and
11 granting permissions beforehand limits performance. In
12 preferred embodiments of the present invention, access is
13 granted in advance based on assumptions regarding the
14 permissions P might need. Checking permissions after the
15 fact does not maintain security. However, such ex post facto
16 checking of permissions allows later checks and audits to be
17 performed by the system. The system may perform such audits
18 periodically at defined intervals. Alternatively, the system
19 may perform the audits during otherwise idle moments.
20 Because audits of this nature can be performed off-line in
21 otherwise idle moments, performance is less impeded.
22 Techniques embodying the present invention are thus less
23 intrusive than conventional techniques. Such audits enable
24 forbidden actions produced by bad grant decisions to be
25 identified. If changes brought about by forbidden actions
26 are recorded, then recovery actions can be taken to return
27 objects to desired states. Audit measures are generally
28 regarded as sufficient for privacy purposes.

29 As indicated earlier, the non-critical class may comprise a
30 plurality of sub classes. For example, in a particularly
31 preferred embodiment of the present invention, there are

1 three classes of actions: 1. informational access control;
2 2. immediate access control; and, 3. deferred access
3 control. Classes 1 and 3 are subclasses of the non-critical
4 class. Class 2 is the critical class.

5 A Class 1 action simply produces an audit record in the
6 access log, but access is always granted. A class 1 action
7 might be, for example, an action to read a publicly
8 available document.

9 A Class 2 action involves prior checking of the access
10 control data and the contents of the access log before it
11 can be executed. A class 2 action is then permitted only if
12 the access control data and the contents of the access log
13 indicate that the permission can be granted. Otherwise, an
14 exception is raised. A class 2 action might, for example, be
15 write operation to a publicly available document.

16 In the case of a Class 3 action, permission need not be
17 checked prior to a grant. Instead, permission is granted and
18 the action is recorded in the access log. The action can
19 then be inspected later, either at a defined interval or
20 during an otherwise idle period, and the quality of the
21 grant decision determined based on the access control data
22 and other accesses recorded in the access log. If the
23 inspection reveals that the access should have not been
24 granted, an alert may be issued. The record of such accesses
25 may include recovery data that enables changes to objects
26 performed downstream of an access allowed via a bad grant
27 decision to be rolled back to an acceptable state. For
28 example, the recovery data may include changes made to a
29 file via addition or deletion, or overwriting of content or

1 example. A class 3 action might for example, be a read from
2 a classified document.

3 It is noted that the present invention is particularly
4 although not exclusively applicable to privacy and data
5 protection. For example, consider a process that accesses,
6 processes, and discloses personal information. To enforce
7 external privacy policy, such disclosures are marked towards
8 outsiders as needing an immediate access control decision.
9 For others, deferred access control might be sufficient.
10 This does not prevent privacy violations within an
11 enterprise, but it prevents such privacy violations
12 producing illegal disclosures of personal information to
13 outsiders.

14 With reference to Figure 2, a data processing system for
15 implementing the present invention comprises a central
16 processing unit (CPU) 200, a memory subsystem 220, an
17 input/output (I/O) subsystem 210, and a bus subsystem 230
18 interconnecting the CPU 200, the memory subsystem 220, and
19 the I/O subsystem 210. Operating system software 240 is
20 stored in the memory subsystem 220. Similarly, at least one
21 object 260 such as a data file is stored in the memory
22 subsystem 220. Access to the object 260 is controlled via
23 access controller software 250 also stored in the memory
24 subsystem 220.

25 Referring now to Figure 3, in operation, the access control
26 software 250 configures the data processing system into
27 logical arrangement in which access to the object 250 by a
28 task 270 executing on the data processing system is
29 controlled by an access controller 280.

1 Referring to Figure 4, on receipt of a request to access the
2 object 250 from the task 270, at block 301, the access
3 controller 280 classifies, at block 302, the request into
4 one of critical and non-critical classes in dependence on
5 stored access control data 285 associated with the object
6 250 and the task 270. If the access is classified into the
7 non-critical class, the access controller 280 grants the
8 task 270 access to the object at block 303 and stores data
9 indicative of the access in an access log 290 at block 304.
10 If the access is classified into the critical class, the
11 access controller 280, at block 305, grants at block 307 or
12 denies at block 306 the task access to the object 250 in
13 dependence on the contents of the access log 290 and the
14 stored access control data 285. The access controller 280
15 may be located in a TCB of the data processing system. As
16 indicated earlier, the TCB is a protected part of the data
17 processing system. In particularly preferred embodiments of
18 the present invention, the TCB may be within a kernel
19 portion of operating system 240.

20 Referring now to Figure 5, in a particularly preferred
21 embodiment of the present invention, in the event that, at
22 block 302, the access is classified into the non-critical
23 class, then, at block 308, the access controller 280
24 determines whether to grant or deny the task 270 access to
25 the object 250 in dependence on the access control data 285.
26 If, at block 308, the access controller 280 decides to grant
27 access at block 303, then the access controller 280 stores a
28 record to this effect is recorded in the access log 290 at
29 block 304. Similarly, if at block 308, the access controller
30 280 decides not to grant access at block 309, then the
31 access controller 280 stores a record to this effect in the
32 access log 290. The simple test performed at block 308 based

1 on the access control data 285 effectively "triages"
2 non-critical access control decisions so that processing
3 power can be focussed instead on more complex decisions
4 based on past event recorded in the access log 290.

5 Referring now to Figure 6 in a preferred embodiment of the
6 present invention, the access controller 280, comprises
7 access control logic 300 for receiving a request to access
8 the object 250 from the task 250. Decision classifier logic
9 310 is connected to the access control logic 300, the access
10 control data 285, and the access log 290 for classifying the
11 access request into one of critical and non-critical classes
12 in dependence on the access control data 285. If the access
13 is classified into the non-critical class, the decision
14 classifier logic 310 grants, the access control logic 300,
15 the task 270 access to the object 250 and stores data
16 indicative of the access in the access log 290. If the task
17 is classified into the critical task, the decision
18 classifier logic passes the request to access control
19 decision logic 320. The access control decision logic 320 is
20 also connected to the access control logic 300, the access
21 log 290, and the access control data 285. On receipt of the
22 critical access request, the access control decision logic
23 320, grants or denies the task 270 access to the object 250
24 in dependence on the contents of the access log 290 and the
25 access control data 285.

26 The non-critical class may be divided into multiple
27 subclasses. Referring now to Figure 7 in a particularly
28 preferred embodiment of the present invention, the access
29 control logic 300 acts as an AEF. Similarly, the decision
30 classification logic 310 acts as a decision triager (ADT)
31 and the access control decision logic 320 acts as an access

1 decision facility (ADF). The access control data 285
2 comprises Access Control Information (ACI) 330 and Access
3 Control Rules (ACR) 360 stored in the memory 220. The ACI
4 330 is substantially as herein before described with
5 reference to Figure 1. In operation, the AEF 300 receives an
6 access request from the task 270. As indicated earlier, the
7 task 270 may be a proxy for a subject in the data processing
8 system, such as a user or a process. The task 270 makes the
9 request because it desires access to the object 250. In
10 response to the request, the AEF 300 generates a decision
11 request. The decision request is routed to the ADT 360. The
12 ADT 310 uses the ACR 360 and ACI 330 to sort the decision
13 request into one of the aforementioned three classes of
14 access; namely:

- 15 1. informational access control;
- 16 2. immediate access control; and,
- 17 3. deferred access control.

18 Here, Class 2 is the critical class. Classes 1 and 3 are
19 subclasses of the non-critical class. The ACI 330 associates
20 the object 290 with a set of access classes. The ACI 330
21 also associates the task 270 with a set of access classes.
22 In typical implementations of access control, the ACR 360
23 and the ACI 330 corresponding to the subject and the object
24 are used to check whether or not access to the object may be
25 granted to the subject. The ACR 360 is divided into two sets
26 of rules. Specifically, the ACR 360 comprises decision rules
27 340 and triage rules 350. The triage rules 340 are used by
28 the ADT 310 in combination with the ACI 330 to classify
29 access requests into one of the aforementioned classes. The
30 decision rules 350 are used by the ADF 320 in combination
31 with the ACI 330.

1 If the ADT 310 assigns the decision request to Class 1 or
2 Class 3, a corresponding default decision is sent from the
3 ADT 310 back to the AEF 300. A corresponding access record
4 is simultaneously stored in the access log 290.

5 If the ADT 310 assigns the decision request to Class 2, then
6 the ADT 310 forwards the decision request to the ADF 320 for
7 further resolution. The ADF 320 uses the contents of the
8 access log 290, the ACI 330, the decision rules 350, and the
9 decision request to arrive at a decision. The ADT 320
10 returns the decision to the AEF 300. The decision may be a
11 grant decision or a signal to raise an exception. The
12 exception decision may additionally trigger recovery
13 actions. Examples of recovery actions will be described
14 shortly.

15 In a particularly preferred embodiment of present invention,
16 the ADT 310 is implemented as a lightweight process and the
17 ADF 320 exerts more effort in arriving at the decision. The
18 ADF 320 may choose to evaluate the contents of the LOG 390
19 without stimulus if, for example, system utilization is low.

20 The ADT 310 can be employed to perform make relatively
21 non-critical decisions herein before described with
22 reference to Figure 5, block 308, leaving the ADF 320 to
23 handle only the more critical decisions. The ADF 320 is not
24 therefore burdened with non-critical activities. Thus,
25 performance of the access controller 280 is greatly
26 improved.

27 In Figure 8, there is shown an example of a privacy access
28 scenario relating to objects in an enterprise. In the

1 scenario, there are two tasks, T1 and T2, operating on three
2 objects O1, O2 and O3. O3 is a publicly accessible resource.
3 Write operations directed to O3 are Class 2, immediate
4 access control, because they have the potential to publicly
5 expose sensitive data. O1 and O2 are both internal resources
6 of the enterprise. Thus, O1 and O2 demand non-critical
7 classification in Classes 1 or 3, deferred and informational
8 access control respectively. Only O1 contains sensitive data
9 such as personal data. T1 and T2 operate unhindered until,
10 at resolution point R, T2 specifies a write operation to O3.
11 At this point, the ADT 310 determines that the attention of
12 the ADF 320 is required. The access rules in this example
13 specify that data exposed publicly, such as that contained
14 in O3, may not be tainted by sensitive data, such as that
15 contained in O1. In addition, the access rules in this
16 example specify that information flows relating to O3 must
17 be examined. In this example, T1 writes to O2 after reading
18 from O1, where sensitive data resides. Thereafter, O2 is
19 potentially tainted by the contents of O1. T2 subsequently
20 reads from potentially tainted O2. Then T2 attempts to write
21 to O3. The ADF 320 detects via the contents of the access
22 log 290 that T2 has read from O2 after T1 has written to O2
23 having previously read from O1. The ADF 320 thus detects
24 that there is potential for O3 to be tainted by sensitive
25 data contained in O1. Accordingly, the ADT 320 determines
26 that access to O3 by T2 should be denied. In a preferred
27 embodiment of the present invention, the ADF 320 raises an
28 exception to prevent further disclosures. In a particularly
29 preferred embodiment of the present invention, T1 and T2 can
30 be rolled back based on stored recovery data so that O2 is
31 no longer potentially tainted by the contents of O1.

1 The present invention permits deferral of access control
2 decisions that may be complex from a computational
3 standpoint to shortly before sensitive information is about
4 to be leaked. This advantageously avoids performing such
5 computations in real-time.

6 Operation of the embodiment of the present invention herein
7 before described with reference to Figure 7 will now
8 described with reference to the flow chart provided in
9 Figure 9.

10 At block 400, an access request arrives at the AEF 300 from
11 the task 270.

12 At block 410 the AEF 300 sends a decision request based on
13 the access request to the ADT 310. On receipt of the
14 decision request, the ADT 310 classifies the access
15 corresponding to the decision request into one of the
16 aforementioned three classes.

17 At block 420, if the access is determined to be in Class 1,
18 informational access control, then, at block 430, a record
19 of the access is saved in the access log 290. At block 440,
20 a decision to grant the access is then sent back to the AEF
21 300 from the ADT 310. If the access is not determined to be
22 in Class 1, then the test at block 450 is performed.

23 At block 450, if the access is determined to be in Class 3,
24 deferred access control, then, at block 460, a record of the
25 access is saved in the access log 290 together with recovery
26 data. Again, at block 440, a decision to grant the access is
27 then sent back to the AEF 300 from the ADT 310. If the
28 access is not determined to be in Class 3, then, at block

1 470, the decision request is forwarded from the ADT 310 to
2 the ADF 320. If the access is not determined to be in Class
3 1 or Class 3, then, by default, the access is determined to
4 be in Class 2, immediate access control.

5 On receipt of the decision request at block 470, the ADF 320
6 evaluates the request based on the access requested, and the
7 contents of the access log 290. If, at block 480, the ADT
8 320 determines from the evaluation that access should be
9 granted, then, at block 440, the ADT 320 issues a decision
10 to this effect to the AEF 300. If, at block 480, the ADT 320
11 determines from the evaluation that access should be denied,
12 then, at block 490, the ADT 320 sends a decision to this
13 effect back to the AEF 300.

14 At block 500, on receipt of a grant decision from the ADF
15 320 and the ADT 310, the AEF 300 grants the task 270 access
16 to the object 250. At block 510, on receipt of a deny
17 decision from the ADF 320, the AEF 300 denies the task 270
18 access to the object 250. In the event that the AEF 300 is
19 in receipt of a deny decision from the ADF 320, additional
20 action may be required, such as aborting the task 270 and
21 raising an exception or rolling back all actions of the task
22 270 and the dependencies of such actions based on stored
23 recovery data.

24 Referring to Figure 10, in another embodiment the present
25 invention, the non-critical class is not subdivided into
26 subclasses. Instead, the test herein before described with
27 reference to Figure 9, block 420 is replaced with test
28 simply to determine whether the access is critical or
29 non-critical. See Figure 10, block 425. If the access is
30 non-critical, then, at block 435, a record of the access is

1 saved in the access log 290 together with recovery data. If
2 the access is critical, then, at block 470, the decision is
3 passed to the ADF 320 as herein before described with
4 reference to Figure 9.

5 As indicated earlier, recovery data may be recorded in the
6 access log 290. The recovery data permits the data
7 processing system to be rolled back to a secure state. In
8 other words, the recovery data permits the data process
9 system to reset itself to the state it enjoyed prior to a
10 bad access grant decision being made. In particularly
11 preferred embodiment of the present invention, the recovery
12 data recorded in the access log 290 comprises change data
13 indicative of changes made to objects when the objects are
14 accessed. Such changes may be additive, such as adding data
15 to files. Alternatively, such changes may be subtractive,
16 such as deleting data from files. The changes include
17 overwriting data in files. It will be appreciated that such
18 changes are generally associated with write operations. In a
19 particularly preferred embodiment of the present invention,
20 each time such changes are made, data indicative of the
21 difference in object content before and after an access was
22 allowed based on a potentially bad grant decision. By
23 recording such difference data, object content prior to the
24 access can be restored in the event that the potentially bad
25 grant decision is determined to be actually bad.

26 Referring to Figure 11, in a preferred embodiment of the
27 present invention, the access log 290 is periodically
28 checked to determine if bad grant decisions have been
29 issued, necessitating remedial action. Specifically, at
30 block 600, a count is checked by the access controller 280.
31 If the count is not reached, then, at block 610, the count

1 is incremented and tested again. If however the count is
2 reached, then, at block 620, the access log 290 is inspected
3 by the ADF 320 to determine, as herein before described with
4 reference Figure 9 blocks 470 and 480, if any bad grant
5 decisions have been issued. If the ADF 320 determines, at
6 block 630, that a bad grant decision has been issued since
7 the last inspection, then, at block 650, the ADT 320 rolls
8 back the affected objects based on the recovery data stored
9 in the access log 290. The access log 290 is then inspected
10 again at block 620 to determine if any other bad grant
11 decisions were made since the last inspection. If the ADT
12 320 determines at block 630 that no bad grant decisions were
13 made since the last inspection, then at block 640, the count
14 is reset, and retested at block 600.

15 Referring to Figure 12, in another preferred embodiment of
16 the present invention, the access log 290 is checked during
17 otherwise idle moments in the data processing system.
18 Specifically, at block 605, the access controller 280 checks
19 the state of the CPU 200. If, at block 615, the access
20 controller 280 determines that the CPU 200, then the check
21 at block 605 is performed again after a predetermined
22 period. If, at block 615, the access controller 280
23 determines that the CPU 200 is free, then blocks 620, 630,
24 and 650 are performed as herein before described with
25 reference to Figure 10. Once all bad grant decisions
26 recorded in the access log 290 since the last inspection
27 have been detected and restoration measures accordingly
28 taken, the test at block 605 is repeated.
29 Preferred embodiments of the present invention have been
30 herein before described with reference to computer program
31 code for configuring the CPU 200 and the memory subsystem
32 220 of a data processing system to perform the functions of

1 the access controller 280, the access control data 285, and
2 the access log 290. It will be appreciated however, that, in
3 other embodiments of the present invention, one or more of
4 such functions may be performed at partially by hardwired
5 logic or similarly dedicated circuitry. Equally, it will be
6 appreciated that the data processing system may be embodied
7 in a single unit or in a plurality of distributed units
8 interconnected via data communications network.

9 In summary, described herein by way of example of the
10 present invention is a method for controlling access to an
11 object in a data processing system comprises: receiving a
12 request to access the object from a task; classifying the
13 access request into one of critical and non-critical classes
14 in dependence on stored access control data associated with
15 the object and the task; granting the task access to the
16 object and storing data indicative of the access in an
17 access log if the access is classified into the non-critical
18 class; and, in the event that the access is classified
19 into the critical class, granting or denying the task access
20 to the object in dependence on the contents of the access
21 log and the stored access control data. It will be
22 appreciated that many implementation of such a method are
23 possible.

24 Variations described for the present invention can be
25 realized in any combination desirable for each particular
26 application. Thus particular limitations, and/or embodiment
27 enhancements described herein, which may have particular
28 advantages to a particular application need not be used for
29 all applications. Also, not all limitations need be
30 implemented in methods, systems and/or apparatus including
31 one or more concepts of the present invention.

1 The present invention can be realized in hardware, software,
2 or a combination of hardware and software. A visualization
3 tool according to the present invention can be realized in a
4 centralized fashion in one computer system, or in a
5 distributed fashion where different elements are spread
6 across several interconnected computer systems. Any kind of
7 computer system - or other apparatus adapted for carrying
8 out the methods and/or functions described herein - is
9 suitable. A typical combination of hardware and software
10 could be a general purpose computer system with a computer
11 program that, when being loaded and executed, controls the
12 computer system such that it carries out the methods
13 described herein. The present invention can also be
14 embedded in a computer program product, which comprises all
15 the features enabling the implementation of the methods
16 described herein, and which - when loaded in a computer
17 system - is able to carry out these methods.

18 Computer program means or computer program in the present
19 context include any expression, in any language, code or
20 notation, of a set of instructions intended to cause a
21 system having an information processing capability to
22 perform a particular function either directly or after
23 conversion to another language, code or notation, and/or
24 reproduction in a different material form.

25 Thus the invention includes an article of manufacture which
26 comprises a computer usable medium having computer readable
27 program code means embodied therein for causing a function
28 described above. The computer readable program code means
29 in the article of manufacture comprises computer readable
30 program code means for causing a computer to effect the

1 steps of a method of this invention. Similarly, the present
2 invention may be implemented as a computer program product
3 comprising a computer usable medium having computer readable
4 program code means embodied therein for causing a a function
5 described above. The computer readable program code means
6 in the computer program product comprising computer readable
7 program code means for causing a computer to effect one or
8 more functions of this invention. Furthermore, the present
9 invention may be implemented as a program storage device
10 readable by machine, tangibly embodying a program of
11 instructions executable by the machine to perform method
12 steps for causing one or more functions of this invention.

13 It is noted that the foregoing has outlined some of the more
14 pertinent objects and embodiments of the present invention.
15 This invention may be used for many applications. Thus,
16 although the description is made for particular arrangements
17 and methods, the intent and concept of the invention is
18 suitable and applicable to other arrangements and
19 applications. It will be clear to those skilled in the art
20 that modifications to the disclosed embodiments can be
21 effected without departing from the spirit and scope of the
22 invention. The described embodiments ought to be construed
23 to be merely illustrative of some of the more prominent
24 features and applications of the invention. Other
25 beneficial results can be realized by applying the disclosed
26 invention in a different manner or modifying the invention
27 in ways known to those familiar with the art.

28 Variations described for the present invention can be
29 realized in any combination desirable for each particular
30 application. Thus particular limitations, and/or embodiment
31 enhancements described herein, which may have particular

1 advantages to the particular application need not be used
2 for all applications. Also, not all limitations need be
3 implemented in methods, systems and/or apparatus including
4 one or more concepts of the present invention.

5 The present invention can be realized in hardware, software,
6 or a combination of hardware and software. A visualization
7 tool according to the present invention can be realized in a
8 centralized fashion in one computer system, or in a
9 distributed fashion where different elements are spread
10 across several interconnected computer systems. Any kind of
11 computer system - or other apparatus adapted for carrying
12 out the methods and/or functions described herein - is
13 suitable. A typical combination of hardware and software
14 could be a general purpose computer system with a computer
15 program that, when being loaded and executed, controls the
16 computer system such that it carries out the methods
17 described herein. The present invention can also be
18 embedded in a computer program product, which comprises all
19 the features enabling the implementation of the methods
20 described herein, and which - when loaded in a computer
21 system - is able to carry out these methods.

22 Computer program means or computer program in the present
23 context include any expression, in any language, code or
24 notation, of a set of instructions intended to cause a
25 system having an information processing capability to
26 perform a particular function either directly or after
27 conversion to another language, code or notation, and/or
28 reproduction in a different material form.

29 Thus the invention includes an article of manufacture which
30 comprises a computer usable medium having computer readable

1 program code means embodied therein for causing a function
2 described above. The computer readable program code means
3 in the article of manufacture comprises computer readable
4 program code means for causing a computer to effect the
5 steps of a method of this invention. Similarly, the present
6 invention may be implemented as a computer program product
7 comprising a computer usable medium having computer readable
8 program code means embodied therein for causing a a function
9 described above. The computer readable program code means
10 in the computer program product comprising computer readable
11 program code means for causing a computer to effect one or
12 more functions of this invention. Furthermore, the present
13 invention may be implemented as a program storage device
14 readable by machine, tangibly embodying a program of
15 instructions executable by the machine to perform method
16 steps for causing one or more functions of this invention.

17 It is noted that the foregoing has outlined some of the more
18 pertinent objects and embodiments of the present invention.
19 This invention may be used for many applications. Thus,
20 although the description is made for particular arrangements
21 and methods, the intent and concept of the invention is
22 suitable and applicable to other arrangements and
23 applications. It will be clear to those skilled in the art
24 that modifications to the disclosed embodiments can be
25 effected without departing from the spirit and scope of the
26 invention. The described embodiments ought to be construed
27 to be merely illustrative of some of the more prominent
28 features and applications of the invention. Other
29 beneficial results can be realized by applying the disclosed
30 invention in a different manner or modifying the invention
31 in ways known to those familiar with the art.